Annales mathématiques Blaise Pascal **Working version – September 19, 2012**

Immersed boundary methods for the numerical simulation of incompressible aerodynamic and fluid-structure interactions

Nicolas James
Emmanuel Maitre
Iraj Mortazavi

Abstract. In this work three branches of Immersed Boundary Methods (IBM) are described and validated for incompressible aerodynamics and fluid-structure interactions. These three approaches are: Cut Cell method, Vortex-Penalization method and Forcing method. Two first techniques are validated for external bluff-body flows and the last one is used for fluid-structure interactions. The paper confirms the ability of this family of numerical schemes for accurate and robust simulation of incompressible flows.

Méthodes de fontière immergée pour la simulation numérique en aérodynamique incompressible et interactions fluide-structure.

Résumé. Dans ce travail, trois méthodes de frontière immergée sont dcrites et validées pour la simulation numérique en aérodynamique incompressible et interactions fluide-structure. Ces trois approches sont : une méthode Cut Cell, une méthode Vortex-Penalisation method et une méthode de forçage. Les deux premières techniques sont valides pour les écoulements autour d'obstacle et la troisième est utilisée pour le calcul d'interactions fluide-structure. Ce papier confirme la capacit de cette famille de schémas numériques pour simuler les écoulements incompressibles de manière précise et robuste.

Keywords: Immersed boundary method, Momentum forcing method, Vortex penalization method, Cut-cell method, Incompressible viscous flows, Complex geometry.
Math. classification: 00X99.

Contents

1. Introduction 2
2. The setting of the problem 5
3. Numerical methods 6
3.1. Cut-cell method 6
3.2. Vortex penalization method 14
3.3. Fluid-Structure interactions with IBM 19
4. Numerical results 24
4.1. Flow around a circular cylinder 24
4.2. Cellular motility and parametric instability 27
5. Conclusions 29
Bibliography 29

1. Introduction

Numerical simulation of fluid mechanics problems is one of the most challenging scientific computing's field of research of the last decades. With the increasing power of computers in terms of memory available, speed and number of processors, more and more complex problems arising in industrial applications become accessible to numerical simulations. In complex geometries, the discretization of the Navier Stokes Equations (NSE) by finite element or finite volume methods on body-fitted grids allows to simulate flows at low to moderate Reynolds numbers. However, generating an efficient conformal mesh is a challenging problem when the geometry gets complex and this pre-processing step is very CPU-time consuming. Indeed, generating a body-fitted grid can be even more expensive than computing the solution itself. Eventhough matrices issued from finite element or finite volume discretizations are dense and sparse, the number of connected nodes is much larger than in finite difference methods on cartesian grids. Specific and more complex solvers have to be used resulting in a significant overhead in the computational effort required to obtain discrete solutions. An alternative to the body-fitted methods is proposed with the Immersed Boundary (IB) methods. The aim of IB methods is to handle complex geometric configurations without the use of body-fitted meshes. Simulations are performed on cartesian grids so that the efficiency and robustness of cartesian grid solvers are achieved. Therefore, significant advances in the

application of numerical simulations to more and more complex industrial problems could be expected.

Since the late 90s, IB methods have been investigated in many published works and among the most investigated problems is the numerical simulations of two or three-dimensional flows past obstacles. Several techniques have been developed to take into account the presence of fixed or mobile obstacles in the flow. In this paper three branches of IB methods will be studied : *Cut-cell methods*, *penalization methods* and *momentum forcing methods*.

The *Cut-cell methods*, called also *Cartesian grid methods*, focus on the discretization of the equations in the mesh cells cut by the immersed boundary (see [50, 48, 13, 32] for instance). In [11, 6], new cut-cell methods, based on the well-known second-order projection MAC scheme [25] are proposed. In order to accurately represent the immersed boundary on the computational cartesian grid, a signed algebraic distance to the obstacle boundary, as in level-set methods [40] is used. The staggered arrangement of the unknowns for both the velocity field and the pressure is adapted to the geometry of the cut-cells. As the boundary conditions are directly imposed, there is no diffusion of the interface fluid-solid. These methods are highly efficient from a computational point of view as it is based on the MAC solver on cartesian grids which has been extensively and successfully used in numerical simulations of turbulent flows, both in the context of direct and Large-Eddy simulations. The only problem with these methods, is the local construction of the discret operators that depends on the position of the immersed boundary in cut-cells.

The second branch, *penalization methods* [1], have a mathematical background. The presence of a solid obstacle in the computational domain is modeled by adding a penalty term, depending on a *penalization* parameter, in the incompressible Navier-Stokes equations. The difference between the penalization solution and the solution of the Navier-Stokes equations can be bounded in Sobolev norms (energy norm for instance) in terms of this parameter. The *penalization method* does not depend on the choice of the discretization schemes used to approximate the equations and is very easy to implement. However, in this approach, the immersed interface is not directly represented in the flow simulation and depends on the mesh properties.

Finally, the third branch covers the IB methods for fluid-structure interactions. They have been first developed by Peskin (see [41], and [42] for a review) for applications in Biology (blood flow). These methods use both Eulerian and Lagrangian variables, which are related via interaction equations. Discrete versions of Dirac functions are used to describe the fluid structure interaction forcing, and to link the Eulerian and Lagrangian variables. In a similar approach, *momentum forcing method*, introduced by Mohd-Yusof [34], have been developped in the context of B-Splines or finite difference methods and applied to the numerical simulations of flows in complex geometries. Here, the challenge is to be able to compute the force applied by the obstacle on the fluid which would be such that the velocity field satisfies the boundary conditions on the immersed boundary (see [46], [51] and [33] for a review). *Momentum forcing methods* are easy to implement and efficient: numerical simulations of flows in complex geometries have been presented in [23] (flows in an full engine) and in [49] (heart valve dynamics). The only drawback with these methods is that, coupling of *momentum forcing methods* with projection schemes introduces difficulties in imposing at the same time level the continuity equation and the boundary conditions on the immersed interface (see [26] for example) and can alter the incompressibility of the flow in the vicinity of the immersed boundary [36].

The advantages and drawbacks of each method are summarized in the Table 1.

The paper is organized as follows: in section 2, we present the problem together with the notation, and we describe the mesh used for the space discretization. In section 3, three numerical methods are detailled: a cut-cell method, a penalty method and a forcing method. Section 4 is then devoted to numerical results for different configurations: the two-dimensional fluid flow around a cylinder at Reynolds number 550 and the study of a three-dimensional vesicle protrusion immersed in a fluid of Reynolds 100. These tests show the robustness of the schemes, comparing them to other experimental and numerical results found in literature.

Methods	Advantages	Drawbacks
Cut-cell	boundary conditions directly imposed	implementation
Penalization	mathematical background, independent of the discretization scheme	parameter dependency
Forcing	physical background, easy to implement	incompressibility constraint

TABLE 1. Advantages and drawbacks of some IB methods.

2. **The setting of the problem**

We consider a two-dimensional flow past a solid obstacle $\Omega^S \subset \mathbb{R}^2$ which is governed by the incompressible Navier-Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} - \nu \Delta \boldsymbol{u} + \boldsymbol{\nabla}(\boldsymbol{u} \otimes \boldsymbol{u}) + \boldsymbol{\nabla} p = 0, \tag{2.1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{2.2}$$

$$\boldsymbol{u}(\boldsymbol{x}, t = 0) = \boldsymbol{u}_0, \tag{2.3}$$

where $\boldsymbol{u}(\boldsymbol{x}, t) = (u, v)$ is the velocity field at location $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$ at time $t > 0$, $\boldsymbol{u}_0$ is the initial condition and $\nu > 0$ is the kinematic viscosity. We assume that the flow fills a rectangular domain $\Omega = (0, L) \times (0, H)$ in which Ω^S is embedded : $\Omega^S \subset \Omega$ (see Figure 1). We denote by Ω^F the fluid domain in which the Navier-Stokes equations (2.1)-(2.3) are prescribed so that we have

$$\Omega = \Omega^F \cup \Omega^S \cup \Gamma^S$$

where $\Gamma^S = \partial\Omega^S$ is the solid boundary. In order to determine the location of each point in the computational domain with respect to the solid boundary Γ^S, we use the signed algebraic distance to Γ^S, which is given by

$$\begin{array}{rccc} d: & \Omega & \longrightarrow & \mathbb{R} \\ & (x, y) & \longmapsto & d(x, y) \end{array} \tag{2.4}$$

and which satisfies : $\Gamma^S = \{(x, y) \in \Omega; d(x, y) = 0\}$. Furthermore, we assume that

$$\begin{cases} d(x, y) > 0 & \text{if } (x, y) \in \Omega^S, \\ d(x, y) < 0 & \text{if } (x, y) \in \Omega^F. \end{cases} \tag{2.5}$$

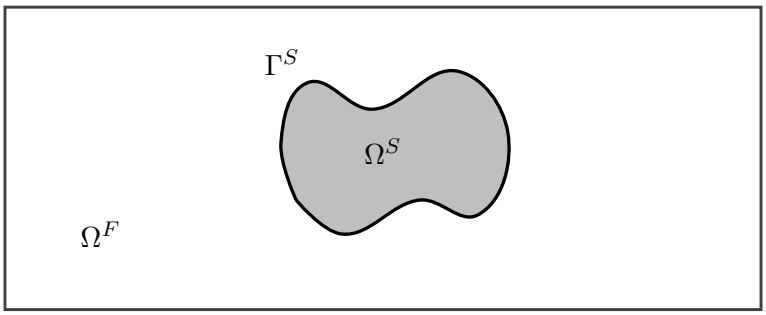

FIGURE 1. The solid body Ω^S with boundary Γ^S and the surrounding computational domain Ω^F in which the flow is to be simulated.

Equations (2.1)-(2.3) are supplemented with boundary conditions on $\Gamma = \partial\Omega$ and on Γ^S. Depending on the problem under consideration, they will be detailed in Section 4. On the immersed boundary Γ^S, Dirichlet boundary conditions are imposed, namely

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) \quad \text{for any } \boldsymbol{x} \in \Gamma^S, \tag{2.6}$$

where $\boldsymbol{g}$ is prescribed. The rectangular computational domain $\Omega = (0, L) \times (0, H)$ is discretized by a cartesian mesh. Let N and M two integers, the sequences of points in each direction satisfy

$$0 = x_0 \leq \ldots \leq x_{i-1} \leq x_i \leq \ldots \leq x_N = L,$$
$$0 = y_0 \leq \ldots \leq y_{j-1} \leq y_j \leq \ldots \leq y_M = H.$$

Mesh sizes are defined by : $\ell_i = x_i - x_{i-1}$ and $h_j = y_j - y_{j-1}$. In order to simplify the notations we denote by $d_{ij} = d(x_i, y_j)$ the algebraic distance of the grid point (x_i, y_j) to the solid boundary Γ^S.

3. Numerical methods

3.1. Cut-cell method

The numerical scheme detailed in this Section is a novel second order *cut-cell method* based on the well-known projection MAC scheme [25], which enforces the incompressibility constraint up to the computer accuracy. In fluid-cells, that is mesh cells which are far enough from the immersed boundary, classical centered, second-order finite volume schemes are used.

In our approach, the location of the velocity component is, as in [11], adapted to the geometry of cut-cells. However, the discrete pressure is placed at the center of the cartesian cells for both fluid-cells and cut-cells. In the vicinity of the obstacle, second-order interpolations using boundary conditions on the solid boundaries are introduced to evaluate the convective fluxes. This results in a local first-order approximation of the nonlinear terms in cut-cells. A pointwise approximation of the viscous terms is used in cut-cells. When boundary conditions on the immersed boundary can be used, a five-point stencil scheme for the viscous term is employed. Otherwise, a six-point first-order approximation is introduced. The resulting linear system is close to the five-point structure symmetric system obtained on cartesian mesh with the MAC scheme. While first-order truncation errors are locally introduced in the scheme in the cut-cells, a second-order global accuracy is recovered. Note that a similar superconvergence result has been proved by Yamamoto in [30] in the context of elliptic equations.

3.1.1. **Time discretization : second-order projection method**

The temporal discretization of (2.1)–(2.3) is achieved by using a second-order projection scheme (BDF2). In a first step, momentum equations are advanced in time with a semi-implicit scheme decoupling the velocity and pressure unknowns. Then, the intermediate velocity is projected in order to obtain a free-divergence velocity field.

Let $\delta t > 0$ stand for the time step and $t^k = k\,\delta t$ discrete time values. Let us consider that $(\boldsymbol{u}^j,\, P^j)$ are known for $j \leq k$. The computation of $(\boldsymbol{u}^{k+1},\, P^{k+1})$ consists in:

– Computing a predictor $\widetilde{\boldsymbol{u}}^{k+1}$ by solving:

$$\begin{aligned}\frac{3\widetilde{\boldsymbol{u}}^{k+1} - 4\boldsymbol{u}^k + \boldsymbol{u}^{k-1}}{2\delta t} - \frac{1}{Re}\,\Delta\widetilde{\boldsymbol{u}}^{k+1} + \boldsymbol{\nabla} P^k = \\ -2\,\boldsymbol{\nabla}\cdot(\boldsymbol{u}^k \otimes \boldsymbol{u}^k) + \boldsymbol{\nabla}\cdot(\boldsymbol{u}^{k-1} \otimes \boldsymbol{u}^{k-1})\end{aligned} \tag{3.1}$$

which is supplemented with Boundary conditions applied to $\widetilde{\boldsymbol{u}}^{k+1}$, depending on the considered problem.

– Projecting to obtain a divergence free velocity $\boldsymbol{u}^{k+1}$:

$$\begin{aligned}&\frac{\boldsymbol{u}^{k+1} - \widetilde{\boldsymbol{u}}^{k+1}}{\delta t} + \frac{2}{3}\,\boldsymbol{\nabla}(P^{k+1} - P^k) = 0,\\ &\boldsymbol{\nabla}\cdot\boldsymbol{u}^{k+1} = 0, \quad (\boldsymbol{u}^{k+1} - \widetilde{\boldsymbol{u}}^{k+1})\cdot\boldsymbol{n} = 0 \quad \text{on } \Gamma.\end{aligned} \tag{3.2}$$

In the following and unless it is necessary, the superscript k denoting discrete times will be omitted.

3.1.2. **The discrete representation of the immersed boundary**

We define by $K_{ij} = (x_{i-1}, x_i) \times (y_{j-1}, y_j)$ the mesh cells in Ω. The horizontal edge $\sigma^x_{i,j}$ of the mesh cell K_{ij} is defined by $\sigma^x_{i,j} = (x_{i-1}, x_i) \times \{y_j\}$, the vertical edge $\sigma^y_{i,j}$ is defined similarly. The computational cells K_{ij}, that is K_{ij} such that $K_{ij} \cap \Omega^F \neq \emptyset$, can be classified in fluid-cells, namely cells which are totally filled by the fluid ($K_{ij} \cap \Omega^S = \emptyset$), and cut-cells, namely cells such that $K_{ij} \cap \Omega^S \neq \emptyset$.

Let a cut-cell K_{ij} and assume that its horizontal edge $\sigma^x_{i,j}$ is cut by the obstacle, namely $\sigma^x_{i,j} \cap \Omega^S \neq \emptyset$ then we compute, as in [11], the ratio r^x_{ij} by

$$r^x_{ij} = \begin{cases} 0 & \text{if } d_{ij} \geq 0 \text{ and } d_{i-1,j} \geq 0, \\ \dfrac{d_{i-1,j}}{d_{i-1,j} - d_{ij}} & \text{if } d_{ij} \geq 0 \text{ and } d_{i-1,j} < 0, \\ \dfrac{d_{i,j}}{d_{ij} - d_{i-1,j}} & \text{if } d_{ij} < 0 \text{ and } d_{i-1,j} \geq 0, \\ 1 & \text{elsewhere.} \end{cases} \tag{3.3}$$

With the ratio r^x_{ij} we approximate the intersection of $\sigma^x_{i,j}$ and Γ^S by the point with coordinates $(x_{i-1} + r^x_{ij}\,\ell_i, y_j)$ if $d_{i-1,j} < 0$ and $(x_i - r^x_{ij}\,\ell_i, y_j)$ if $d_{i,j} < 0$. Similarly, we defined the intersection of $\sigma^y_{i,j}$ and Γ^S. By doing so, the obstacle boundary Γ^S is approximated by a piecewise-linear curve Γ^S_h (see Figure 2). This provides a discrete separation of the computational domain in $\Omega = \Omega^F_h \cup \Omega^S_h \cup \Gamma^S_h$.

Let a computational cell K_{ij}, that is $K_{ij} \cap \Omega^F_h \neq \emptyset$, we denote by K^F_{ij} the part of the cell which is filled by the fluid, namely $K^F_{ij} = K_{ij} \cap \Omega^F_h$ and by $\sigma^{x,F}_{i,j}$ (resp. $\sigma^{y,F}_{i,j}$) the part of the edge $\sigma^x_{i,j}$ (resp. $\sigma^y_{i,j}$) which lies in Ω^F_h, so that we have $\sigma^{x,F}_{i,j} = \sigma^x_{i,j}$ if K_{ij} is a fluid-cell and

$$\sigma^{x,F}_{i,j} = \begin{cases} (x_{i-1}, x_{i-1} + r^x_{ij}\,\ell_i) \times \{y_j\} & \text{if} \quad d_{i-1,j} < 0, \\ (x_i - r^x_{ij}\,\ell_i, x_i) \times \{y_j\} & \text{if} \quad d_{i,j} < 0, \end{cases}$$

if K_{ij} is a cut-cell.

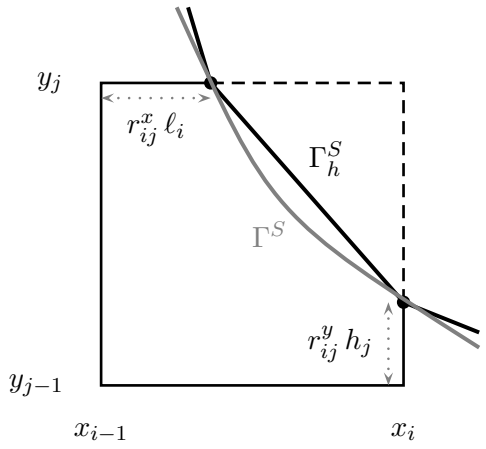

FIGURE 2. The discrete representation Γ_h^S of the immersed boundary Γ^S.

3.1.3. **A staggered arrangement of the unknowns**

As in the classical MAC scheme for cartesian grids (see [25]), the velocity unknowns are located at the midpoints of the cell edges. Let a computational cell K_{ij}^F, the associated velocity unknowns u_{ij} and v_{ij} are respectively located at the midpoints $\boldsymbol{\kappa}_{ij}^{y,F}$ of $\sigma_{ij}^{y,F}$ and $\boldsymbol{\kappa}_{ij}^{x,F}$ of $\sigma_{ij}^{x,F}$. Note that

$$\boldsymbol{\kappa}_{ij}^{y,F} = \begin{cases} (x_i, y_{j-1/2}) & \text{if} \quad \sigma_{ij}^{y,F} \subset \Omega_h^F, \\ (x_i, y_{j-1} + \frac{r_{ij}^y h_j}{2}) & \text{if } \sigma_{ij}^y \cap \Omega_h^S \neq \emptyset \text{ and } d_{i,j-1} < 0, \\ (x_i, y_j - \frac{r_{ij}^y h_j}{2}) & \text{if } \sigma_{ij}^y \cap \Omega_h^S \neq \emptyset \text{ and } d_{i,j} < 0, \end{cases}$$

where $y_{j-1/2} = \frac{1}{2}(y_j + y_{j-1})$. Even if $K_{i,j}^F$ is a cut-cell, the discrete pressure value P_{ij} is always located at the center of the corresponding mesh cell $K_{i,j}$, that is at point of coordinates $(x_{i-1/2}, y_{j-1/2})$. This staggered arrangement of the unknowns is represented on Figure 3.

3.1.4. **Description of the discrete operators**

In this section, we describe the numerical discretization of the spatial partial derivatives present in the Navier-Stokes equations. Specific numerical

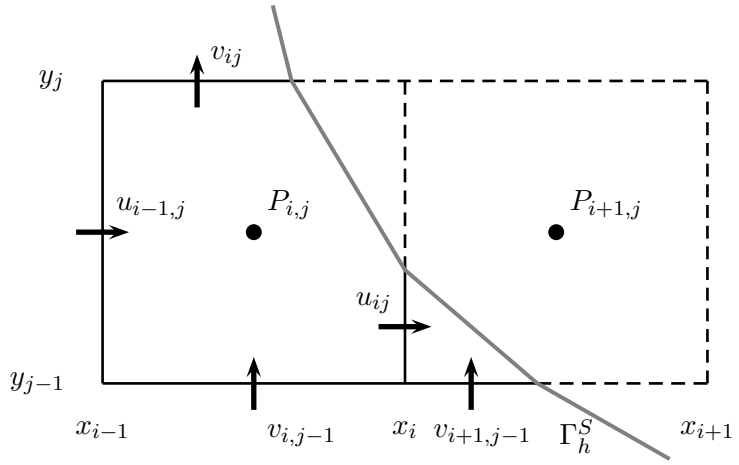

FIGURE 3. Location of the unknowns in the cut-cells K_{ij} (left) and $K_{i+1,j}$ (right).

treatments are required in cut-cells while, away from the obstacle, classical second-order centered finite volume schemes are used. As the discret operators depend on the local geometrical configuration of the cut-cells, we refer the reader to [6] for more details about the method.

The Laplace operator. In the classical MAC scheme on cartesian grid, the horizontal (resp. vertical) momentum equation is discretized by integration over the volume cell $K_{ij}^u = (x_{i-1/2}, x_{i+1/2}) \times (y_{j-1}, y_j)$ (resp. $K_{ij}^v = (x_{i-1}, x_i) \times (y_{j-1/2}, y_{j+1/2})$). In order to take into account the presence of the obstacle, integration is performed over the computational cells $K_{ij}^{u,F} = K_{ij}^u \cap \Omega_h^F$ and $K_{ij}^{v,F} = K_{ij}^v \cap \Omega_h^F$. In the following, we restrict the presentation to the equation of the horizontal velocity component. The vertical case is treated similarly.

Let a computational volume control $K_{ij}^{u,F}$, which is surrounding the velocity unknown u_{ij}, the approximation of $\int_{K_{i,j}^{u,F}} \Delta u \, d\mathbf{x}$ depends on the location of the mesh cell $K_{i,j}$ with respect to the obstacle. If $K_{i,j}$ is sufficiently far from the obstacle so that both $K_{i,j}$ and $K_{i+1,j}$ are fluid cells, that is $(K_{i,j} \cup K_{i+1,j}) \subset \Omega_h^F$, and the faces $\sigma_{i,j-1}^y$ and $\sigma_{i,j+1}^y$ are not cut by the obstacle, then the following classical second-order approximation is

used

$$\begin{aligned}\frac{1}{|K_{i,j}^{u,F}|}\int_{K_{i,j}^{u}}\Delta u\,d\mathbf{x} &= \frac{1}{|K_{i,j}^{u,F}|}\int_{\partial K_{i,j}^{u}}\frac{\partial u}{\partial n}\,ds\\ &\approx \frac{2}{(\ell_{i+1}+\ell_i)}\left(\frac{u_{i+1,j}-u_{i,j}}{\ell_{i+1}}-\frac{u_{i,j}-u_{i-1,j}}{\ell_i}\right)\\ &+\frac{2}{(h_{j+1}+h_j)}\left(\frac{u_{i,j+1}-u_{i,j}}{h_{j+1}}-\frac{u_{i,j}-u_{i,j-1}}{h_j}\right).\end{aligned}\tag{3.4}$$

In the other cases, that is either $K_{i,j}$, $K_{i+1,j}$, $\sigma_{i,j-1}^{y}$ or $\sigma_{i,j+1}^{y}$ are cut by the obstacle, we use a pointwise approximation, namely

$$\frac{1}{|K_{i,j}^{u,F}|}\int_{K_{i,j}^{u,F}}\Delta u\,d\mathbf{x}\;\approx\;\Delta u(\boldsymbol{\kappa}_{i,j}^{y,F}).$$

Obviously such approximation is first-order. When it is feasible, depending upon the location of $\boldsymbol{\kappa}_{i,j}^{y,F}$ with respect to the obstacle, the Dirichlet condition on Γ_h^S is used to write a five-point stencil, first-order finite difference approximation of $\Delta u(\boldsymbol{\kappa}_{i,j}^{y,F})$. Otherwise, a sixth point chosen in the neighbourhood of $\boldsymbol{\kappa}_{i,j}^{y,F}$, namely $\{\boldsymbol{\kappa}_{i\pm1,j\pm1}^{y,F}\}$, is added in the approximation scheme. Therefore, for mesh cells close to the obstacle, a locally first-order error is done. We observe (Taylor-Couette flow test case) that this locally first-order truncation error near the obstacle does not affect the global second-order convergence rate, note that the same behaviour was observed in [28]. We mention that a similar superconvergence result has also been proved by Yamamoto in [30] in the context of elliptic equations.

The nonlinear (convective) terms. Let a computational volume control $K_{ij}^{u,F}$, which is surrounding the velocity unknown u_{ij}, the nonlinear term is discretized by first writing

$$\int_{K_{ij}^{u,F}}\boldsymbol{\nabla}(u\boldsymbol{u})\,d\boldsymbol{x}\;=\;\int_{\partial K_{ij}^{u,F}}u\,\boldsymbol{u}\cdot\boldsymbol{n}\,ds\;=\;\sum_{\sigma\in\partial K_{ij}^{u,F}}\int_{\sigma}u\,\boldsymbol{u}\cdot\boldsymbol{n}\,ds\tag{3.5}$$

where $\boldsymbol{n}$ denotes the unit normal vector to the boundary $\partial K_{ij}^{u,F}$, outward to $K_{ij}^{u,F}$. The computational cell $K_{i,j}^{u,F}$ is a polygon which is bounded by, at most, six edges. In (3.5), the numerical approximation of $\int_\sigma u\,\boldsymbol{u}\cdot\boldsymbol{n}\,ds$

is done by using the midpoint integration rule, which induces a third-order error. For an edge shared with the obstacle boundary, the velocity boundary conditions are directly used. On the other edges, an interpolation is required in order to obtain an approximation of the velocity at the midpoint edges. A second-order interpolation is used and the interpolation formulae depend on how the cells are cut by the obstacle. Let us denote by $F^{\sigma}_{i,j}$ the discrete flux corresponding to an edge $\sigma \in \partial K^{u,F}_{i,j}$. Assuming that the velocity field $\boldsymbol{u}$ is sufficiently regular, we show that the discretization error near the obstacle is first order. Elsewhere, a second order approximation is achieved.

The pressure gradient. Let a computational volume control $K^{u,F}_{ij}$. The approximation of the mean pressure gradient over $K^{u,F}_{i,j}$ depends on the location of the mesh cell $K_{i,j}$ with respect to the obstacle. If $K_{i,j}$ is sufficiently far from the obstacle so that both $K_{i,j}$ and $K_{i+1,j}$ are fluid cells, that is $(K_{i,j} \cup K_{i+1,j}) \subset \Omega^F_h$, then the following classical second-order approximation is used

$$\frac{1}{|K^u_{i,j}|}\int_{K^u_{i,j}} \frac{\partial P}{\partial x}\, d\mathbf{x} = \frac{1}{|K^u_{i,j}|}\int_{\partial K^u_{i,j}} P\, n_x\, ds \approx 2\,\frac{(P_{i+1,j} - P_{i,j})}{(\ell_i + \ell_{i+1})}. \tag{3.6}$$

If either $K_{i,j}$ or $K_{i+1,j}$ are cut-cell, a first order pointwise approximation is applied at $\boldsymbol{x} = \boldsymbol{\kappa}^{y,F}_{i,j}$. Combining the standard finite difference approximation with a second-order interpolation, we define the discrete horizontal component of the pressure gradient at point $\boldsymbol{\kappa}^{y,F}_{i,j}$ by

$$(G_x P)_{i,j} = \begin{cases} (1 + r^y_{ij})\frac{P_{i+1,j} - P_{i,j}}{\ell_i + \ell_{i+1}} + (1 - r^y_{ij})\frac{P_{i+1,j-1} - P_{i,j-1}}{\ell_i + \ell_{i+1}} \\ \qquad\qquad \text{if } d_{i,j-1} < 0 \text{ and } d_{i,j} > 0, \\ (1 + r^y_{ij})\frac{P_{i+1,j} - P_{i,j}}{\ell_i + \ell_{i+1}} + (1 - r^y_{ij})\frac{P_{i+1,j+1} - P_{i,j+1}}{\ell_i + \ell_{i+1}} \\ \qquad\qquad \text{if } d_{i,j-1} > 0 \text{ and } d_{i,j} < 0, \\ 2\frac{(P_{i+1,j} - P_{i,j})}{(\ell_i + \ell_{i+1})} \text{ elsewhere.} \end{cases} \tag{3.7}$$

The vertical component of the pressure gradient is discretized similarly at point $\boldsymbol{\kappa}^{x,F}_{i,j}$ and, its discrete approximation is denoted by $(G_y P)_{i,j}$.

The continuity equation. As in the classical MAC scheme on cartesian grids, the discrete divergence is obtained by integration over a computational cell K_{ij}^F, namely we write

$$\int_{K_{ij}^F} \mathrm{div}(\boldsymbol{u})\, d\boldsymbol{x} \;=\; \sum_{\sigma\in\partial K_{ij}^F} \int_{\sigma} \boldsymbol{u}\cdot\boldsymbol{n}\, ds, \tag{3.8}$$

where $\boldsymbol{n}$ denotes the unit normal vector to the boundary ∂K_{ij}^F, outward to K_{ij}^F. The computational cell $K_{i,j}^F$ is a polygon which is bounded by at most five edges With the use of the midpoint integration rule in (3.8), we derive the discrete version of the continuity equation in the cell $K_{i,j}^F$, namely

$$(D\boldsymbol{u})_{i,j} \;+\; |\sigma_{i,j}^S|\, \boldsymbol{u}(\boldsymbol{\kappa}_{i,j}^S)\cdot\boldsymbol{n}(\boldsymbol{\kappa}_{i,j}^S) \;=\; 0. \tag{3.9}$$

where $(D\boldsymbol{u})_{i,j} \;=\; \ell_i(r_{i,j}^x v_{i,j} - r_{i,j-1}^x v_{i,j-1}) \;+\; h_j(r_{i,j}^y u_{i,j} - r_{i-1,j}^y u_{i-1,j})$.

The velocity correction step. In projection schemes, the intermediate velocity $\tilde{\boldsymbol{u}}$ is corrected with the help of the pressure gradient in order to obtain a free-divergence velocity field. For the horizontal velocity, the correction step is discretized by integration of (3.2) over each edges $\sigma_{ij}^{y,F}$, namely we write

$$\int_{\sigma_{ij}^{y,F}} \Big(u - \tilde{u} + \frac{2}{3}\delta t \frac{\partial}{\partial x}(\delta P^{k+1})\Big)\, dy \;=\; 0 \tag{3.10}$$

where $\delta P^{k+1} = P^{k+1} - P^k$. The discrete version of (3.10) follows by writing

$$u_{ij} - \tilde{u}_{ij} + \frac{2}{3}\delta t (G_x \delta P^{k+1})_{i,j} \;=\; 0 \tag{3.11}$$

and similarly for the vertical velocity component

$$v_{ij} - \tilde{v}_{ij} + \frac{2}{3}\delta t (G_y \delta P^{k+1})_{i,j} \;=\; 0. \tag{3.12}$$

By reporting (3.11) and (3.12) in the definition of $(D\boldsymbol{u})_{ij}$ above, and using (3.9), we derive

$$\begin{aligned} &\ell_i r_{i,j}^x\, (G_y \delta P^{k+1})_{i,j} - \ell_i r_{i,j-1}^x\, (G_y \delta P^{k+1})_{i,j-1} \;+\; h_j r_{i,j}^y\, (G_x \delta P^{k+1})_{i,j} \\ -h_j r_{i-1,j}^y\, (G_x \delta P^{k+1})_{i-1,j} \;&=\; \frac{3}{2\delta t}(D\tilde{\boldsymbol{u}})_{i,j} \;+\; \frac{3}{2\delta t}\, |\sigma_{i,j}^S|\, \boldsymbol{u}(\boldsymbol{\kappa}_{i,j}^S)\cdot\boldsymbol{n}(\boldsymbol{\kappa}_{i,j}^S). \end{aligned} \tag{3.13}$$

which is a discrete Poisson equation satisfied by the pressure increment δP^{k+1} in a computational cut-cell $K_{i,j}^F$. As for the classical MAC scheme, the resolution of this linear system ensures that the incompressiblity condition is enforced up to the computer accuracy.

3.2. **Vortex penalization method**

Vortex methods (see [18], [22] and [35]) and penalization methods (see [1] and [9]) have been separately used to compute incompressible high Reynolds number flows around obstacles. In this section, a novel hybrid particle-penalization technique is proposed to achieve efficient computations of bluff-body flows designing a more efficient technique that covers the advantages of both approaches [17]. In this approach, the vortex method is used to approximate the penalized Vorticity Transport Equations (VTE). This technique that permits to solve the flow equations in a fast lagrangian way, overcomes the difficulty of the vortex methods to satisfy accurately the no-slip boundary conditions, introducing the penalization term in the Vorticity Transport Equations. Here, the idea is to extend the fluid velocity inside the solid body and to solve the flow equations with a penalization term to enforce rigid motion inside the solid, using a vorticity formulation. The main interest of the penalized vorticity formulation is that it replaces the usual vorticity creation algorithm in order to satisfy the no-slip boundary condition for vortex methods. This new technique avoids the convergence difficulties due to the creation of the particles on the solid boundaries. This approach is also able to take into account the moving obstacles and boundaries in the flow thanks to an immersed boundary algorithm that is used to complement this hybrid technique as shown in [14] and [15]. Then, the method is validated for two-dimensional transitional and turbulent flows around a moving vertical axis turbine.

3.2.1. **Vortex method**

Taking the curl of the momentum equation (2.1), we get the Helmholtz or the Vorticity Transport Equation (VTE):

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega - \omega \cdot \nabla \mathbf{u} = \nu \, \Delta \omega \quad \text{in } \Omega, \tag{3.14}$$

expressing the transport of vorticity by convection and diffusion. Here, $\mathbf{u}$ is the velocity vector and ω is the vorticity (curl of the velocity). Solving this equation permits us to convect and diffuse the vorticity field. Once the vorticity field known, integrating the vorticity and the continuity equation (2.2) permits to describe the velocity field. The unsteady structure of these equations permits to find out the solution along the time. Finally, because the VTE doesn't contain directly the pressure term (taking the curl of Navier-Stokes equations this term is vanished), the pressure can be recovered from the velocity field.

The previous equations are approximated using a Vortex method [18]. These methods that are very robust and low-cost to simulate high Reynolds number recirculating flows, are more recently used to implement control techniques in fluid mechanics [22]. In this kind of approach, the VTE equation (3.14) is solved using a two-fractional step (or viscous splitting) method. It corresponds to approximate separately the diffusion and convection terms at each time step. The two fractional steps are:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega - \omega \cdot \nabla \mathbf{u} = 0, \tag{3.15}$$

$$\frac{\partial \omega}{\partial t} = \nu \, \Delta \omega. \tag{3.16}$$

The convective part is solved using a "Vortex-In-Cell (VIC)" method (see e.g. [18] or [22]) with a semi-Lagrangian resolution. In this fractional step, the transport of vorticity due to convection is obtained from the solution of equation (3.15) in terms of the Lagrangian displacement of a set of finite vortex elements. If we define the stream function ψ by:

$$u = \frac{\partial \psi}{\partial y} \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x}, \tag{3.17}$$

and then substituting it in the vorticity definition $\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$, the following Poisson equation is achieved:

$$-\Delta \psi = \omega. \tag{3.18}$$

Let assume that a number nv^n of the finite vortex elements, located at $\mathbf{x}^n(it)$ with a circulation (strength) $\gamma^n(it)$, are known at time t^n, $1 \leq it \leq nv^n$. Let also assume that the vorticity field $\omega^n(x_i, y_j)$ at time t^n is known on the Cartesian mesh. The main target of the resolution method is to compute the same quantities at time $t^{n+1} = t^n + \delta t$ as nv^{n+1}, $\mathbf{x}^{n+1}(it)$, $\gamma^{n+1}(it)$ and $\omega^{n+1}(x_i, y_j)$.

In the first step of the computational procedure, the equation (3.18) with associated boundary conditions is solved to recover the stream function field $\psi^n(x_i, y_j)$. Then, solving the equation (3.17), the velocity $\mathbf{u}^n(x_i, y_j)$ is computed at each node of the mesh. Finally, using a high order interpolation procedure, a convective velocity $\mathbf{u}_v^n(it)$ is associated to each finite vortex element ($1 \leq it \leq nv^n$), and the convective displacement is achieved using a fourth order Runge-Kutta method. Particles are then remeshed on the original grid. Finally diffusion is solved through a implicit solver on the grid, with a classical 7-points second order scheme.

Then, the final location of each finite vortex element $\mathbf{x}_*^{n+1}(it)$ at time t^{n+1} is obtained as the sum of the convective and diffusive movements during the last time step:

$$\mathbf{x}_*^{n+1}(it) = \mathbf{x}^n(it) + \mathbf{dl}_{conv}^n(it) + \mathbf{dl}_{diff}^n(it). \tag{3.19}$$

In incompressible flows the unique source of generation of the vorticity is the no-slip boundary condition. The no-slip boundary condition is expressed in terms of the velocity field at the wall and does not involve explicitly the vorticity. To overcome this problem, classical vortex methods that directly deal with the vortex elements, mimic the vortex generation phenomenon on solid boundaries. So, the slip velocity on the wall is nullified by the generation of new vortex elements on the boundary [12]; it also compensates the vorticity leaving the domain from the exit boundary. Nevertheless, there is no straightforward mathematical proof related to the convergence of this strategy for Navier-Stokes wall-bounded flows. The alternative to this approach is to satisfy the no-slip conditions using the penalization method that does not need any local condition on the solid-fluid interface. In this work, this later approach is coupled to the vortex method avoiding the almost heuristic vortex generation approach.

3.2.2. **Penalization method for the velocity formulation**

The penalization method was initially designed to be introduced in the Navier-Stokes equations in order to take into the account solid obstacles in fluid flows. In this section, we show how the penalization method can be used successfully to model the flow of an incompressible fluid around an obstacle [1]. In the penalization technique the system is considered as a single flow, subject to the Navier-Stokes equation with a penalization

term that enforces continuity at the solid-fluid interface and rigid motion inside the solid. We solve simultaneously the Brinkman equations in the solid and the Navier-Stokes equations in the fluid, considering whole the domain as a porous medium with zero (solid) or infinite permeabilities (fluid). The main advantage of this method is that it needs neither the mesh to fit the boundaries nor to specify no-slip boundary conditions. In addition it allows to compute the pressure as a continuous field on the whole domain including the solids, and the lift and drag coefficients by integrating the penalization term inside the solid bodies [9].
The zone variation is realized changing the penalization coefficient that defines the permeability of each region. Numerically, the fluid is considered as a porous medium with a very high permeability ($K = 10^{16}$) and the bodies are considered as porous media with a very small permeability ($K = 10^{-8}$) . Let us define a penalization parameter $\lambda \approx 1/K$, that is $\lambda \to 0$ in the fluid region Ω^F and $\lambda \gg 1$ in the solid region Ω^S. By means of the λ, the velocity term is penalized for a solid in Brinkman equations. That means that Navier-Stokes equations are replaced by the following equations:

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} + \lambda \mathbf{u} + \nabla p = 0 \quad in\ \Omega \tag{3.20}$$

$$div\ \mathbf{u} = 0 \quad in\ \Omega \tag{3.21}$$

where λ is the penalization parameter with the dimension $[s^{-1}]$.

3.2.3. **Penalization method for vorticity formulation**

In this section, the idea is to extend the fluid velocity inside the solid body and to solve the flow equations with a penalization term to enforce rigid motion inside the solid, using a vorticity formulation. The main interest of the penalized vorticity formulation is that it replaces the usual vorticity creation algorithm in order to satisfy the no-slip boundary condition for vortex methods. This new technique avoids the convergence difficulties due to the particle creation on the solid boundaries (see[15] and [19]). Defining the Reynolds number as $\mathrm{Re} = u_{ref}\ l_{ref}/\nu$, the non-dimensional penalized vorticity equation reads:

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\mathbf{u} + \frac{1}{\mathrm{Re}} \Delta \omega + \lambda \nabla \times [\chi_S(\bar{\mathbf{u}} - \mathbf{u})], \tag{3.22}$$

where χ_S is the characteristic function that yields 0 in the fluid and 1 in the solid and $\bar{\mathbf{u}}$ indicates the velocity of the solid body.

To discretize the penalized vorticity equation (3.22) in a vortex method, the equation is split in three substeps. At each time step, one successively solves the following equations:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \lambda \boldsymbol{\nabla} \times [\chi_S(\overline{\mathbf{u}} - \mathbf{u})] \tag{3.23}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\, \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\mathbf{u} + \frac{1}{Re}\Delta \boldsymbol{\omega} \tag{3.24}$$

To solve (3.23) we use a an implicit scheme [15] and we set:

$$\widetilde{\boldsymbol{\omega}}^{n+1} = \boldsymbol{\nabla} \times \left[\frac{u^n + \lambda \Delta t \chi_S \overline{\mathbf{u}}^n}{1 + \lambda \Delta t \chi_S} \right]. \tag{3.25}$$

where Δt is the time step. The right hand side above is evaluated by centered finite differences.
At this stage, grid vorticity above a certain cut-off is used to create particle at grid point locations and equation (3.24) is solved by a classical vortex-in-cell method [18]. The velocity field is obtained solving:

$$\Delta \psi = -\boldsymbol{\omega} \tag{3.26}$$

with boundary conditions on the stream function ψ where $\mathbf{u} = \nabla \times \psi$. Particles are pushed with a RK4 time-stepping. Particles are then remeshed on the original grid using the following third order interpolation kernel

$$\Lambda_3(x) = \begin{cases} 0 & \text{if } |x| > 2 \\ \frac{1}{2}(2 - |x|)^2(1 - |x|) & \text{if } 1 \leq |x| \leq 2 \\ 1 - \frac{5x^2}{2} + \frac{3|x|^3}{2} & \text{if } |x| \leq 1 \end{cases} \tag{3.27}$$

Finally diffusion is solved through an implicit solver on the grid, with a classical 7-points second order scheme. Note that the same kernel is used to interpolate grid velocity values onto particles in the RK4 particle pusher. Grid values for vorticity, velocity and level set functions are now available for time t_{n+1} and a new cycle of iterations can start. Moreover, the no-slip boundary conditions are naturally satisfied penalizing the vorticity transport equations.

3.3. **Fluid-Structure interactions with IBM**

The Immersed Boundar Methods were originally designed for fluid-structure computations in biology and are progressed towards more robust techniques more recently. We will start from a Lagrangian representation of the membrane since it is more usual, and then show that an Eulerian formulation based on the level set method is equivalent.

3.3.1. **Lagrangian elasticity of an immersed interface**

Let a smooth elastic surface $\widetilde{\Gamma}$ in $\mathbb{R}^3$ in a rest configuration, parametrized by a regular $\theta : [0, M]^2 \to \mathbb{R}^3$, $M > 0$. We assume that this surface reacts only to membrane deformation (no flexural effects), and more specifically to area change, and not to membrane shear. As well, we considered a surface with a boundary which is not coherent with a vesicle, but this is to introduce the Lagrangian representation. This will be worked around automatically by the Level Set representation. For the sake of simplicity we assume the membrane to react only to tangential stress and more specifically to area change. The membrane surface density in that configuration is denoted by $\lambda_\theta(r, s)$. The surface is moving between time $t = 0$ et $t = T$, and we call Γ_t, its position at time t. In particular Γ_0 is its initial position, and generally $\Gamma_0 \neq \widetilde{\Gamma}$ unless the elastic surface is initially at rest. We denote by $(r, s) \to \gamma_0(r, s)$ and $(r, s) \to \lambda_0(r, s)$ a regular parametrization and surface density for Γ_0 such that $\lambda_0|\gamma_{0,r} \times \gamma_{0,s}| = \lambda_\theta|\theta_r \times \theta_s|$. Let $\gamma : [0, M]^2 \times [0, T] \to \gamma(r, s, t)$ the regular parametrization of Γ_t transported by the velocity field u of the continuous medium, that is $\gamma(r, s, t) = X(t; \gamma_0(r, s))$ or equivalently:

$$\begin{cases} \partial_t \gamma(r, s, t) = u(\gamma(r, s, t), t), & (r, s) \in [0, M]^2, t \in]0, T] \\ \gamma(r, s, 0) = \gamma_0(r, s), & (r, s) \in [0, M]^2. \end{cases} \tag{3.28}$$

Γ_t is immersed into a Newtonian incompressible and homogeneous fluid with given density ρ_f and visocity μ. This example corresponds to consider a singular density for the whole continuous medium defined by:

$$\rho = \rho_f + \lambda \delta_{\Gamma_t}$$

where δ_{Γ_t} is the measure supported by Γ_t, which is defined by

$$\forall h \in \mathcal{C}_0^0(\Omega), \qquad \langle \delta_{\Gamma_t}, h \rangle = \int_{\Gamma_t} h(x) d\sigma.$$

3.3.2. **Immersed boundary model**

Formally, we obtain a non-homogeneous Navier-Stokes equation with a singular forcing term and a density with a singular part. We write it under the following **compact form of IBM:**

$$\begin{array}{ll}
(\rho_f + \lambda\delta_{\Gamma_t})(\partial_t u + u\cdot\nabla u) - 2\boldsymbol{\nabla}\cdot\mu D(u) + \nabla p & \\
\quad = F(r,s,t)\delta_{\Gamma_t} + (\rho_f + \lambda\delta_{\Gamma_t})g & \text{in } \Omega\times]0,T[\\
\boldsymbol{\nabla}\cdot u = 0 & \text{on } \Omega\times]0,T[\\
\partial_t\gamma = u(\gamma,t) & \text{on } [0,M]\times]0,T[
\end{array}$$

where $F(r,s,t)$ and g are respectively the elastic force per unit of surface and the gravity force. In addition, the evolution of the elastic surface is dictated by (3.28).

This formulation is exactly the immersed boundary method from Peskin [43, 42] although it is written in a different form, and for an elastic surface rather than a volume collection of fibers. Indeed using Peskin's notations the immersed boundary condition amounts to mix Eulerian and Lagrangian quantities. Fluid unknowns are Eulerian while Lagrangian markers are used for the surface. The interaction of these two representations is done thanks to a discrete Dirac measure. With the not so mathematical notations of [42], adapted to the surface case (terms in $|\theta_r \times \theta_s|$), **IBM form of Peskin** reads:
Eulerian description of velocity field and *Lagrangian description* of immersed structure (made of a family of elastic fibers), interpolated in the Eulerian domain.

- A velocity field $(x,t) \in \Omega \times [0,T] \to u(x,t)$.
- $(r,s,t) \in U \times [0,T] \to \gamma(r,s,t)$ position of points on the elastic surface Γ_t.
- Force density with respect to the surface measure (r,s) in the reference configuration is a known function $F_\theta(r,s,t)$.
- The surface density in the reference configuration is a known function $\lambda_\theta(r,s,t)$.

- Equations of motion (stress coupling):

$$(\rho_f + \Lambda)(\partial_t u + u \cdot \nabla u) - \nu \Delta u + \nabla p = f \tag{3.29}$$

$$\boldsymbol{\nabla} \cdot u = 0 \tag{3.30}$$

$$f(x,t) = \int_U |\theta_r \times \theta_s| F_\theta(r,s,t) \delta(x - \gamma(r,s,t)) dr ds \tag{3.31}$$

$$\Lambda(x,t) = \int_U |\theta_r \times \theta_s| \lambda_\theta(r,s,t) \delta(x - \gamma(r,s,t)) dr ds \tag{3.32}$$

$$\partial_t \gamma = u(\gamma(r,s,t),t) = \int_\Omega u(x,t) \delta(x - \gamma(r,s,t)) dx \tag{3.33}$$

$$F_\theta(r,s,t) = \mathcal{F}_\theta[\gamma(r,s,t)] \tag{3.34}$$

The immersed boundary method we just described is simple and attractive. However, at each time iteration one has to convert back and forth the coordinates, which introduces serious volume conservation issues. Indeed the interpolated velocity field is not divergence free thus the curve described does not enclose a volume of constant measure. This volume loss is acknowledged and in a good extent cured in [44, 29, 27], but the method looses its inherent simplicity. The foremost aim of our Eulerian formulation, introduced in [19, 20] was to maintain the method simplicity while introducing an Eulerian localization of the interface which suppress these interpolation problems. The original immersed boundary method can be implemented so that it is order 2 in the case of thick interfaces, but order 1 for thin structures [24]. Stability studies have been developed in [3, 4, 47].

Let us point out that our Eulerian formulation will by structure make natural the handling of closed membrane, which is less straightforward in Lagrangian coordinates. Moreover, the variable viscosity or density case, which is not considered in the IBM is very simply handled in our formulation. This is important in application to biological cell models that often exhibit a viscosity contrast between the inner and outer fluids to take into account the biological material inside it.

3.3.3. **Eulerian Elasticity of an immersed membrane**

We now skip to a new representation of the interface to avoid caveats encountered with the Lagrangian formulation.

Level Set formulation. We assume that Γ_t is the zero level set of a function $\phi : \Omega \times [0,T] \to \mathbb{R}$ so that

$$\Gamma_t = \{x \in \Omega, \quad \phi(x,t) = 0\}.$$

As $\phi(\gamma(r,s,t),t) = 0$ on $[0,M]^2 \times [0,T]$, and $\partial_t \gamma = u(\gamma,t)$, we get by time differentiation

$$\partial_t \phi(\gamma(r,s,t),t) + u(\gamma(r,s,t),t) \cdot \nabla \phi(\gamma(r,s,t),t) = 0.$$

The Level Set method [39] takes as initial condition a function ϕ_0 whose zero level set is Γ_0 and amounts to find a function ϕ which is solution to the above transport equation on the whole computational domain:

$$\begin{cases} \partial_t \phi + u \cdot \nabla \phi = 0 & \text{on } \Omega \times]0,T[\\ \phi = \phi_0 & \text{on } \Omega \times \{0\}. \end{cases} \tag{3.35}$$

A common choice for ϕ_0 is the signed distance to the interface:

$$\phi_0(x) = \begin{cases} -\operatorname{dist}(x,\Gamma_0) & \text{if } x \text{ is inside } \Gamma_0, \\ \operatorname{dist}(x,\Gamma_0) & \text{if } x \text{ is outside } \Gamma_0. \end{cases}$$

With this choice of ϕ_0 the exterior normal to the domain enclosed by Γ_t, and the surface mean curvature are expressed in terms of ϕ as follows:

$$n(x) = \frac{\nabla \phi}{|\nabla \phi|} \qquad \kappa(x) = \boldsymbol{\nabla} \cdot \frac{\nabla \phi}{|\nabla \phi|}$$

Deformations and Level Set. What is more original and proved in [19] is that in the case of an incompressible flow, the surface stretching is recorded in the function ϕ, which allows us to rephrase our fluid-structure problem with this function. There is several ways to prove this intuitive fact, see [8, 7, 2]. We could start from the stretching written in Lagrangian and prove that $|\boldsymbol{\nabla}\phi|$ verifies the same equation, as in [19] that is to say

$$\partial_t |\boldsymbol{\nabla}\phi| + u \nabla |\boldsymbol{\nabla}\phi| = -|\boldsymbol{\nabla}\phi| \frac{\boldsymbol{\nabla}\phi^T \nabla u \boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|^2} = -|\boldsymbol{\nabla}\phi| \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|} \otimes \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|} : \nabla u \tag{3.36}$$

However we will propose a more intrinsic demonstration, which relies on the following proposition.

Proposition 3.1. *Let $u : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ of class $\mathcal{C}^1$ with $\boldsymbol{\nabla} \cdot u = 0$ and ϕ solution of class $\mathcal{C}^1$ of $\partial_t \phi + u \cdot \nabla \phi = 0$, $\phi = \phi_0$ with $|\nabla \phi| \geq \alpha > 0$ in a neighborhood of $\{\phi = 0\}$. We assume that $s \to \int_{\{|\phi(x,t)|<s\}} f(x)dx$ is*

of class $\mathcal{C}^1$ in a neighborhood of $s = 0$ for any $t \in [0, T]$ and $f \in \mathcal{C}_c(\mathbb{R}^n)$. Then for every function f continuous and with compact support,

$$\int_{\{\phi_0(\xi)=0\}} f(\xi)|\nabla\phi_0|^{-1}(\xi)d\sigma(\xi) = \int_{\{\phi(x,t)=0\}} f(Y(x,t))|\nabla\phi|^{-1}(x,t)d\sigma(x) \tag{3.37}$$

which means that $|\nabla\phi|/|\nabla\phi_0|$ represents the variation of surface measure between Γ_t and Γ_0.

Corollary 3.2. *In dimension 3, if $(r, s) \in \omega \to \gamma(r, s, t) \in \mathbb{R}^3$ is a (patch of) parametrization of Γ_t, we still have $Y(\gamma(r, s, t), t) = \gamma(r, s, 0)$ which gives*

$$\int_\omega f(\gamma(r,s,0))|\nabla\phi_0|^{-1}(\gamma(r,s,0))|\gamma_r \times \gamma_s|(r,s,0)drds$$
$$= \int_\omega f(\gamma(r,s,0))|\nabla\phi|^{-1}(\gamma(r,s,t),t)|\gamma_r \times \gamma_s|(r,s,t)drds$$

therefore

$$\frac{|\nabla\phi|(\gamma(r,s,t),t)}{|\nabla\phi_0|(\gamma(r,s,0))} = \frac{|\gamma_r \times \gamma_s|(r,s,t)}{|\gamma_r \times \gamma_s|(r,s,0)}.$$

More precisely we construct ϕ_0 such that its zero level set is Γ_0, and such that

$$|\nabla\phi_0|(\gamma(r,s,0)) = \frac{|\gamma_r \times \gamma_s|(r,s,0)}{|\theta_r \times \theta_s|(r,s)}$$

which corresponds to the area change between the initial and rest configuration. Then we have

$$|\nabla\phi|(\gamma(r,s,t),t) = \frac{|\gamma_r \times \gamma_s|(r,s,t)}{|\theta_r \times \theta_s|(r,s)}.$$

If the initial stretching does not depend on (r, s) (uniform stretching) this amounts to initialize ϕ_0 to this stretching times the signed distance to the surface.

Energy and elastic force expressed in Level Set. We are now in a position to express the elastic energy of an elastic membrane in terms of the Level Set function. But we must first cope with the surface integral, which leads us to consider the following energy (here for a surface with

stretching 1 at rest)

$$\mathcal{E}[\phi] = \int_{\{\phi=0\}} E(|\boldsymbol{\nabla}\phi|)\frac{1}{|\boldsymbol{\nabla}\phi|}d\sigma.$$

Then we could differentiate this energy with respect to t in order to identify the singular elastic force (Thomas Milcent in his thesis [31] inspected this direction). We could then develop a finite element method to give a weak meaning to measure, which is made in [3, 4] in the framework of immersed boundary method. However we chose, as in the usual Level Set method, to approximate the surface measure by a non singular function. Then we can use a finite difference method on a cartesian grid with fast FFT solver for Poisson equation. In this aim we have the following Proposition (proved in [20] and originally in [10]) which gives a volume approximation of a Dirac Measure supported by an hypersurface localized by a level set:

Proposition 3.3. *Let $r \to \zeta(r)$ be a continuous function with support in $[-1, 1]$, such that $r \to \frac{1}{\varepsilon}\zeta(\frac{r}{\varepsilon})$ converges toward δ_0 in the sense of distributions. Then under assumption (H_ϕ), when $\varepsilon \to 0$,*

$$\frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)|\nabla\phi| \rightharpoonup \delta_{\{\phi=0\}} \quad \text{in } \mathcal{M}(\mathbb{R}^d).$$

Therefore we see that a sound numerical approximation of $\delta_{\{\phi=0\}}$ is given by $|\nabla\phi|\frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)$, which allows to define a regularized energy by

$$\mathcal{E}_\varepsilon(\phi) = \int_\Omega E(|\boldsymbol{\nabla}\phi|)\frac{1}{\varepsilon}\zeta(\frac{\phi}{\varepsilon})d\mathbf{x}. \tag{3.38}$$

4. Numerical results

4.1. Flow around a circular cylinder

In this section the *cut-cell* and the *vortex-penalization* methods are used to perform computations around a circular cylinder. Here, to take into account a transitional case we focus on an impulsively started flow at Re = 550 and compare the results to [45]. The time evolution of drag coefficients are studied using the 'momentum equation' method ([37]), as shown on the Figure 4. It should be outlined that the 'hydrodynamical impulse' method needs a zero far-field velocity. Here, a Galilean transformation is used moving the body with a $-u_\infty$ velocity to achieve the correct boundary

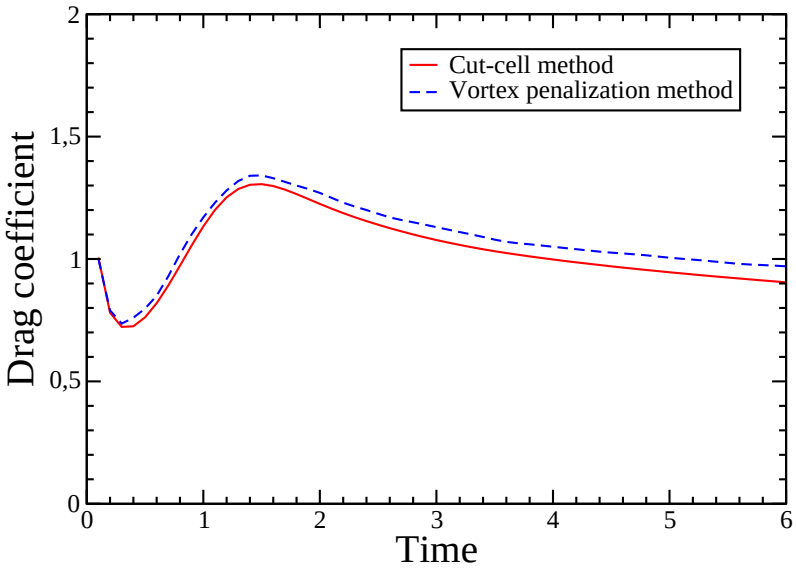

FIGURE 4. Comparison of drag evolution for cut-cell and penalization methods for an impulsively started flow around a circular cylinder at Reynolds 550.

and force computations. Moreover, in order to avoid any reflecting effect of the outgoing vortices from the exit boundaries the computations are performed on a short time $t = 6$. As the curve shows, the results have a good agreement. Also, they correspond to the drag computations obtained in [45] (almost placed just between two curves). The figure 5 shows the vorticity iso-contours for both methods which are very similar to each other and to the vorticity field presented by Ploumhans & Winckelmans [45], introducing accurately the separation contours and the recirculation area sizes. Here, the grid convergence in a computational domain $[-3.75,\ 12.5]\times$

FIGURE 5. Vorticity contours around an impulsively started flow around a circular cylinder for times 1, 3 and 5 (from top to bottom) at Reynolds number 550 (left: vortex-penalization method; right: cut-cell method)

$[-3.75,\ 3.75]$ is achieved with parameters $h = 0.005$ and $\lambda = 10^9$ for the penalization method.

4.2. **Cellular motility and parametric instability**

Several tests have been developed in order to validate the fluid-structure *forcing* method [20, 21]. We present an application of our method to the study of vesicle protrusions. Cells, for example for their motion, create protrusions on their membrane. The underlying mechanism is still controversial in the Biologists community. It is attributed to polymerization / depolymerization process of actin filaments in the neighborhood of the membrane, but this is modeled differently between authors (cf [38, 5] and references therein), either formulated as a two-phase continuous medium, a visco-elastic medium, or as a microscale network. In an article of Cortez, Peskin, Stockie and Varela [16] the authors study the parametric instability of the system formed by an elastic membrane immersed in a $2D$ fluid. Parametric instability is well known to everybody who once use a swing. In the immersed membrane case, we could imagine that the periodic variation of somme parameter of the system, as the stiffness for example, could lead to large unstable deformations. This is what is proved in [16] and that we study here numerically in dimension 3.

From a numerical point of view the tests we developed with our membrane model clearly illustrated that a spherical membrane slightly perturbed, and with a stiffness varying with a precise periodicity, could exhibit an instability, which means very large displacements compared with the initial perturbation. We want to stress out that this does not correspond to a resonance phenomenon under a suitably chosen forcing term. For example without initial perturbation of the membrane the periodic fluctuations of its stiffness do not create any motion.

The following example (Figure 6) is academic: it corresponds to a membrane with stiffness 1 immersed in a fluid of Reynolds 100. The spherical membrane is perturbed in the following way: we consider a meridian on this sphere that we perturb by a small amplitude (2.5%) oscillation in $\cos(4\theta)$. Then we consider the surface generated by this perturbed meridian. The membrane stiffness is oscillating between 0.5 and 1.5. The pictures below show the deformed immersed membrane (the fluid is not represented). The colors on the surface give the stretching. The pressure slices in the middle of the vesicle are plotted on the edges of the graphical box.

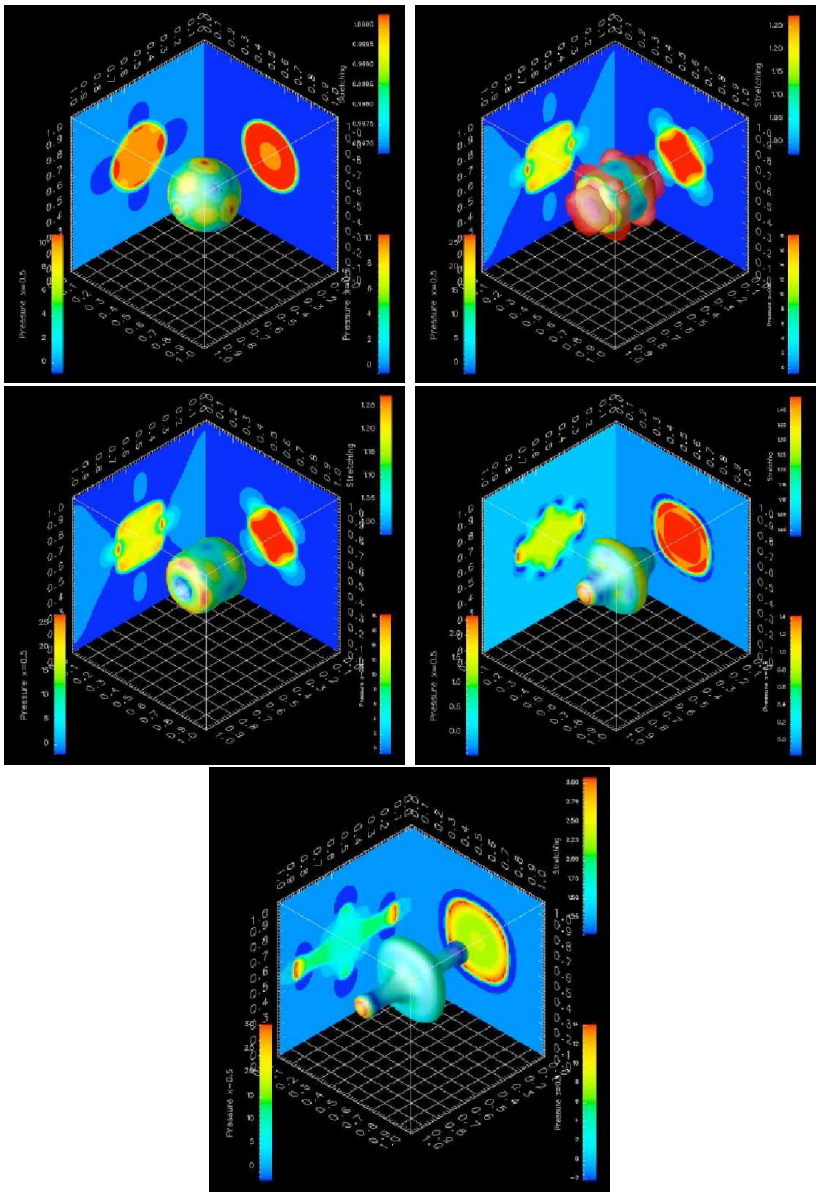

FIGURE 6. Parametric instability of an immersed elastic membrane

5. Conclusions

In this paper, three branches of Immersed Boundary Methods (IBM) were analyzed and validated for incompressible aerodynamics (*Cut-Cell* and *vortex penalization* techniques) and fluid-structure interactions (*Forcing* method). All three methods show that the IBM either with Eulerian or with Lagrangian frameworks can efficiently simulate incompressible flows with fixed and moving boundaries. The paper also underlines some challenging problems to increase the robustness of these promising approaches.

Bibliography

[1] P. Angot, C. H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.*, 81:497–520, 1999.

[2] J. T. Beale and J. Strain. Locally corrected semi-lagrangian methods for stokes flow with moving elastic interfaces. *J. Comp. Phys.*, 227(8):3896–3920, 2008.

[3] D. Boffi, L. Gastaldi, and L. Heltai. Numerical stability of the finite element immersed boundary method. *M3AS*, 17:1479–1505, 2007.

[4] D. Boffia, L. Gastaldi, and L. Heltai. Stability results and algorithmic strategies for the finite element approach to the immersed boundary method, preprint available on http//www.ing.unibs.it/˜gastaldi/paper.html. *Proceeding of the Sixth European Conference on Numerical Mathematics and Advanced Applications*, pages 557–566, 2005.

[5] S. Bohnet, R. Ananthakrishnan, A. Mogilner, J.-J. Meister, and A. Verkhovsky. Weak force stalls protrusion at the leading edge of the lamellipodium. *Biophys. J.*, 90:1810–1820, 2006.

[6] F. Bouchon, T. Dubois, and N. James. A second-order cut-cell method for the numerical simulation of 2d flows past obstacles. *Computers and Fluids*, 65:80–91, 2012.

[7] D. Bresch, T. Colin, E. Grenier, B. Ribba, and O. Saut. Computational modeling of solid tumor growth: the avascular stage. *SIAM Journal on Scientific Computing*, 32(4):2321–2344, 2010.

[8] D. Bresch, Th. Colin, E. Grenier, B. Ribba, O. Saut, O. Singh, and C. Verdier. Quelques méthodes de paramètre d'ordre avec applications à la modélisation de processus cancéreux. *ESAIM: Proceedings*, 18:163–180, 2007.

[9] Ch. H. Bruneau, I. Mortazavi, and P. Gilliéron. Passive control around the two-dimensional square back ahmed body using porous devices. *J. Fluids Eng.*, 130, 2008.

[10] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *J. Comp. Phys.*, 124:449–464, 1996.

[11] Y. Cheny and O. Botella. The ls-stag method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. *J. Comp. phys.*, 229:1043–1076, 2010.

[12] A.J. Chorin. Vortex sheet approximation of boundary layers. *J. Comput. Phys.*, 27, 1978.

[13] M.-H. Chung. Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape. *Computers and Fluids*, 35(6):607–623, 2006.

[14] M. Coquerelle, J. Allard, G. H. Cottet, and M. P. Cani. A Vortex Method for Bi-phasic Fluids Interacting with Rigid Bodies. *Arxiv preprint math, LMC-IMAG*, 2006.

[15] M. Coquerelle and G. H. Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys.*, 227, 2008.

[16] R. Cortez, C.S. Peskin, J.M. Stockie, and D. Varela. Parametric resonance in immersed elastic boundaries. *SIAM Journal on Applied Mathematics*, 65(2):494–520, 2004.

[17] G. H. Cottet, F. Gallizio, A. Magni, and I. Mortazavi. A vortex immersed boundary method for bluff body flows. *ASME Summer Meeting, Montreal*, FEDSM-ICNMM2010-30787, 2010.

[18] G. H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice*. 2000.

[19] G.-H. Cottet and E. Maitre. A level-set formulation of immersed boundary methods for fluid-structure interaction problems. *C. R. Math.*, 338(7):581–586, 2004.

[20] G.-H. Cottet and E. Maitre. A level set method for fluid-structure interactions with immersed surfaces. *Math. Models Meth. Appl. Sci.*, 16(3):415–438, 2006.

[21] G.-H. Cottet, E. Maitre, and T. Milcent. Eulerian formulation and level set models for incompressible fluid-structure interaction. *ESAIM-Math. Model. Numer. Anal.*, 42:471–492, 2008.

[22] E. Creusé, A. Giovannini, and I. Mortazavi. Vortex simulation of active control strategies for transitional backward-facing step flows. *Computers & Fluids*, 38, 2009.

[23] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite difference methods for three-dimensional complex flow simulations. *J. Comput. Phys.*, 161:35–60, 2000.

[24] B.E. Griffith and C.S. Peskin. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *J. Comp. Phys.*, 208:75–105, 2005.

[25] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 12:2182–2189, 1965.

[26] J. Kim, D. Kim, and H. Choi. An immersed-boundary finite volume method for simulation of flow in complex geometries. *J. Comput. Phys.*, 171:132–150, 2001.

[27] L. Lee and R.J. Leveque. An immersed interface method for incompresible navier-stokes equations. *SIAM J. Sci. Comp.*, 25(3):832–856, 2003.

[28] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.

[29] R. J. LeVeque and Z. Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18(3):709–735, 1997.

[30] N. Matsunaga and Y. Yamamoto. Superconvergence of the shortley-weller approximation for dirichlet problems. *J. Comp. Appl. Math.*, 116:263–273, 2000.

[31] T. Milcent. *Formulation eulerienne du couplage fluide structure, analyse mathématique et applications en biomécanique.* Thèse de l'Université de Grenoble, nov 2008.

[32] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, and A. V. Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys.*, 227:4825–4852, 2008.

[33] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[34] J. Mohd-Yusof. *Combined immersed-boundary/B-Spline methods for simulations of flow in complex geometries*, pages 317–327. NASA Ames Research Center/Stanford University, 1997.

[35] I. Mortazavi and A. Giovannini. The simulation of vortex dynamics downstream of a plate separator using a vortex-finite element method. *Int. J. Fluid Dynamics*, 5, 2001.

[36] F. Muldoon and S. Acharya. A divergence-free interpolation scheme for the immersed boundary method. *Int. J. Numer. Method Fluid*, 56:1845–1884, 2008.

[37] F. Noca, D. Shiels, and D. Jeon. A comparison of methods for evaluating time-dependent fluid dynamic forces on bodies, using only velocity fields and their derivatives. *Journal of Fluids and Structures*, 13, 1999.

[38] D. Olz, C. Schmeiser, and V. Small. Modelling of the actin-cytoskeleton in symmetric lamellipodial fragments. *Cell Adhesion and Migration*, 2(2):117–126, 2008.

[39] S. Osher and R. P. Fedkiw. *Level set methods and Dynamic Implicit Surfaces*. Springer, 2003.

[40] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[41] C. S. Peskin. The fluid dynamics of heart valves: experimental, theoretical, and computational methods. *Ann. Rev. Fluid Mech.*, 14:235–259, 1982.

[42] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:1–39, 2002.

[43] C.S. Peskin. Numerical analysis of blood flow in the heart. *J. Comp. Phys.*, 25:220–252, 1977.

[44] S. Peskin and B.F. Printz. Improved volume conservation in the computation of flows with immersed boundaries. *J. Comput. Phys.*, 105:33–46, 1993.

[45] P. Ploumhans and G. S. Winckelmans. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *Journal of Computational Physics*, 165, 2000.

[46] E. M. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *J. Comput. Phys.*, 123:450–465, 1996.

[47] J. Stockie. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. *J. Comp. Phys.*, 154:41–64, 1999.

[48] P. G. Tucker and Z. Pan. A cartesian cut-cell method for incompressible viscous flow. *Appl. Math. Model.*, 24:591–606, 2000.

[49] M. De Tullio, A. Cristallo, E. Balaras, G. Pascazio, P. De Palma, G. Iaccarino, M. Napolitano, and R. Verzicco. Recent advances in the immersed boundary method. In P. Wesseling, E. Oñate, and J. Périaux, editors, *ECCOMAS CFD*, 2006.

[50] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *J. Comp. Phys.*, 156:209–240, 1999.

[51] N. Zhang and Z. C. Zheng. An improved direct-forcing immersed boundary method for finite difference applications. *J. Comput. Phys.*, 221:250–268, 2007.

Nicolas James
LMA Université de Poitiers
UMR CNRS 7348
Téléport 2 - BP 30179
Bd Marie et Pierre Curie
86962 Chasseneuil FRANCE
nicolas.james@math.univ-poitiers.fr

Emmanuel Maitre
LJK Université de Grenoble
UMR CNRS 5224
Tour IRMA, BP 53
51, rue des Mathématiques
38041 Grenoble Cedex 9 FRANCE
Emmanuel.Maitre@imag.fr

Iraj Mortazavi
IMB Université de Bordeaux
UMR CNRS 5251
MC^2 INRIA Bordeaux Sud-Ouest
351, cours de la libération
33405 Talence FRANCE
mortaz@math.u-bordeaux1.fr